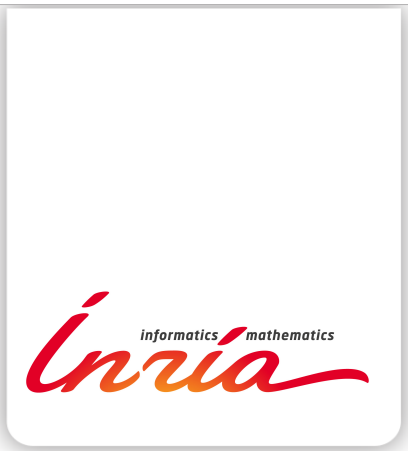
A Sensitivity Equation Method for Unsteady Compressible Flows: Implementation and Verification

Régis Duvigneau

RESEARCH REPORT
N° 8739
June 2015
Project-Team Acumes

ISSN 0249-6399 ISRN INRIA/RR--8739--FR+ENG



A Sensitivity Equation Method for Unsteady Compressible Flows: Implementation and Verification

Régis Duvigneau

Project-Team Acumes

Research Report n° 8739 — June 2015 — 34 pages

Abstract: Sensitivity analysis is a key element in a design optimization procedure. Although the related theory and numerical implementation are well known for steady problems, the application to unsteady partial differential equations, in particular in fluid mechanics, is still an active research area.
In this report, a sensitivity equation method is described, in the context of compressible Navier-Stokes equations, and an efficient numerical implementation is proposed. The resulting approach is verified for two- and three-dimensional problems of increasing complexity.

Key-words: sensitivity analysis, unsteady flows

RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Une méthode à équations de sensibilités pour les écoulements compressibles instationnaires : implémentation et vérification

Résumé : L'analyse de sensibilité est un élément clé dans une procédure de conception optimale. Bien que la théorie et l'implémentation numérique sous-jacente soient bien connues pour les problèmes stationnaires, l'application aux équations aux dérivées partielles instationnaires, en particulier en mécanique des fluides, est encore un domaine de recherche actif.

Dans ce rapport, on décrit une méthode à équations de sensibilités, dans le contexte des équations de Navier-Stokes compressible, et on propose une implémentation numérique efficace. L"approche résultante est vérifée pour des problèmes bi- et tri-dimensionnels de complexité croissante.

Mots-clés : analyse de sensibilité, écoulements instationnaires

Introduction

Sensitivity analysis is now a common tool in design optimization procedures. In the context of aerodynamic design, this task is usually achieved by solving adjoint equations. The main advantage of such a method is the independency of the computational cost with respect to the number of design parameters. Sensitivity analysis can be done on the basis of a continuous approach (differentiate-then-discretize) or discrete approach (discretize-then-differentiate), which have been developed successively for steady Euler equations first, and turbulent Navier-Stokes then [7, 12, 13, 15, 17, 18]. Note that Automatic Differentiation (AD) tools can provide a significant help for the discrete case [8, 16, 19].

Nowadays, realistic flow simulations rely more and more on unsteady analyses, because of the use of sophisticated turbulence modeling, such as Detached-Eddy Simulations (DES) or Large-Eddy Simulations (LES) which are intrinsically unsteady, or because the problem studied has an unsteady definition (moving bodies, time-dependent boundary conditions, etc). However, the extension of the adjoint method to the unsteady context is far from being straightforward. Indeed, this approach requires a forward time integration of the state equations and then a backward time integration of the adjoint equations. Therefore, the storage of all variable fields at all time steps is necessary. This is obviously a serious difficulty when considering DES or LES computations, which require very long simulation time intervals for statistical analyses. To alleviate this difficulty, some strategies based on partial storage and partial re-computations have been proposed recently [2, 9].

The objective of the current work is to study an alternative approach, based on the sensitivity equation method. Contrary to the adjoint equations, sensitivity equations are integrated forward in time and do not need a complete storage of the solution. Nevertheless, their computational cost depends on the number of parameters considered. Some works can be found in the literature on the sensitivity equation method for incompressible flow problems, for different applications like optimum design, uncertainty propagation or flow characterization [3, 6, 11, 21, 22]. However, only a few works can be found related to compressible flows.

In this report, the implementation of the method is described in the context of unsteady compressible Navier-Stokes equations, for laminar and subsonic flow regimes. A particular attention is paid to the ease of implementation in an existing simulation code, and to the computational efficiency. Several verification exercises are demonstrated for two- and three-dimensional, steady and unsteady problems.

1 Flow equations

For the sake of generality, we consider the two-dimensional Favre-averaged Navier-Stokes equations, that can be written in the conservative form as follows:

$$\frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \vec{\mathcal{F}} = \nabla \cdot \vec{\mathcal{G}}, \tag{1}$$

where $\mathbf{W}$ are the conservative mean flow variables $(\rho, \rho u, \rho v, \rho e)$, with ρ the density, $\vec{U} = (u, v)$ the velocity vector and e the total energy per unit of mass. $\vec{\mathcal{F}} = (\mathbf{F}_x(\mathbf{W}), \mathbf{F}_y(\mathbf{W}))$ is the vector of the convective fluxes and $\vec{\mathcal{G}} = (\mathbf{G}_x(\mathbf{W}), \mathbf{G}_y(\mathbf{W}))$ the vector of the diffusive fluxes. The pressure p is obtained from the perfect gas state equation:

$$p = \rho(\gamma - 1)(e - \frac{1}{2}\|\overrightarrow{U}\|^2) = \rho(\gamma - 1)e_i \tag{2}$$

where γ is the ratio of the specific heat coefficients and e_i the internal energy. We denote $\mathbf{U}$ the primitive flow variables (ρ, u, v, p). The extension to the three-dimensional case is straightforward.

The inviscid fluxes are given by:

$$\mathbf{F}_x(\mathbf{W}) = \begin{pmatrix} \rho u \\ \rho u^2 + p \\ \rho uv \\ \rho u(e + \frac{p}{\rho}) \end{pmatrix} \quad \mathbf{F}_y(\mathbf{W}) = \begin{pmatrix} \rho v \\ \rho vu \\ \rho v^2 + p \\ \rho v(e + \frac{p}{\rho}) \end{pmatrix}. \tag{3}$$

The viscous fluxes are written as:

$$\mathbf{G}_x(\mathbf{W}) = \begin{pmatrix} 0 \\ \tau_{xx} \\ \tau_{yx} \\ u\tau_{xx} + v\tau_{yx} - q_x \end{pmatrix} \quad \mathbf{G}_y(\mathbf{W}) = \begin{pmatrix} 0 \\ \tau_{xy} \\ \tau_{yy} \\ u\tau_{xy} + v\tau_{yy} - q_y \end{pmatrix}, \tag{4}$$

where the symmetric viscous stress tensor $\bar{\bar{\tau}}$ is defined as:

$$\begin{aligned} \tau_{xx} &= \frac{2}{3}(\mu + \mu_t)\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right), \\ \tau_{xy} &= (\mu + \mu_t)\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \\ \tau_{yy} &= \frac{2}{3}(\mu + \mu_t)\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right), \end{aligned} \tag{5}$$

and the heat flux $\vec{q}$ based on Fourier's law:

$$(q_x, q_y) = -\lambda(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t})\left(\frac{\partial e_i}{\partial x}, \frac{\partial e_i}{\partial y}\right). \tag{6}$$

μ and λ are the viscosity and thermal conduction coefficients, and Pr is the Prandtl number. μ_t and Pr_t correspond to turbulent viscosity and Prandtl numbers.

The boundary conditions at the wall correspond to adiabatic non slipping conditions, written as:

$$\vec{U}(\vec{x}_w) = \vec{0} \qquad (\vec{q} \cdot \vec{n})(\vec{x}_w) = 0. \tag{7}$$

At far field boundaries, fluxes are prescribed:

$$\left(\vec{\mathcal{F}} \cdot \vec{n}\right)(\vec{x}_{far}) = \mathcal{F}_0 \qquad \left(\vec{\mathcal{G}} \cdot \vec{n}\right)(\vec{x}_{far}) = \mathcal{G}_0 \tag{8}$$

2 Sensitivity equations

We consider in this section the differentiation of the previous flow equations with respect to a parameter, denoted a, that could be either scalar or geometrical. a could be a control parameter for design optimization purpose, or an uncertain parameter for uncertainty quantification. The objective of this section is to derive a set of partial differential equations governing the sensitivity fields denoted:

$$\mathbf{W}' = \frac{\partial \mathbf{W}}{\partial a}. \tag{9}$$

Differentiating (1) with respect to a yields:

$$\frac{\partial}{\partial a}\left(\frac{\partial \mathbf{W}}{\partial t}\right) + \frac{\partial}{\partial a}\left(\nabla \cdot \vec{\mathcal{F}}\right) = \frac{\partial}{\partial a}\left(\nabla \cdot \vec{\mathcal{G}}\right). \tag{10}$$

By switching the derivatives with respect to a and those with respect to time or space coordinates, one obtains:

$$\frac{\partial}{\partial t}\left(\frac{\partial \mathbf{W}}{\partial a}\right) + \nabla \cdot \left(\frac{\partial \vec{\mathcal{F}}}{\partial a}\right) = \nabla \cdot \left(\frac{\partial \vec{\mathcal{G}}}{\partial a}\right), \tag{11}$$

or:

$$\frac{\partial \mathbf{W}'}{\partial t} + \nabla \cdot \vec{\mathcal{F}}' = \nabla \cdot \vec{\mathcal{G}}', \tag{12}$$

which is formally similar to (1), by introducing the sensitivity of the convective flux $\vec{\mathcal{F}}' = (\mathbf{F}'_x(\mathbf{W}, \mathbf{W}'), \mathbf{F}'_y(\mathbf{W}, \mathbf{W}'))$ and the sensitivity of the diffusive flux $\vec{\mathcal{G}}' = (\mathbf{G}'_x(\mathbf{W}, \mathbf{W}'), \mathbf{G}'_y(\mathbf{W}, \mathbf{W}'))$. This remark is especially important, since one intends to re-use existing numerical methods developed to solve the flow problem (1), to solve the sensitivity problem (12).

The sensitivity of the convective fluxes can be expressed as:

$$\mathbf{F}'_x(\mathbf{W}, \mathbf{W}') = \begin{pmatrix} (\rho u)' \\ (\rho u)'u + (\rho u)u' + p' \\ (\rho u)'v + (\rho u)v' \\ (\rho u)'(e + \frac{p}{\rho}) + (\rho u)(e' + (\frac{p}{\rho})') \end{pmatrix} \tag{13}$$

$$\mathbf{F}'_y(\mathbf{W}, \mathbf{W}') = \begin{pmatrix} (\rho v)' \\ (\rho v)'u + (\rho v)u' \\ (\rho v)'v + (\rho v)v' + p' \\ (\rho v)'(e + \frac{p}{\rho}) + (\rho v)(e' + (\frac{p}{\rho})') \end{pmatrix}. \tag{14}$$

This particular form will be useful when a finite-volume formulation will be adopted.

The sensitivity of the diffusive fluxes reads:

$$\mathbf{G}'_x(\mathbf{W},\mathbf{W}') = \begin{pmatrix} 0 \\ \tau'_{xx} \\ \tau'_{yx} \\ u'\tau_{xx} + v'\tau_{yx} + u\tau'_{xx} + v\tau'_{yx} - q'_x \end{pmatrix} \tag{15}$$

$$\mathbf{G}'_y(\mathbf{W},\mathbf{W}') = \begin{pmatrix} 0 \\ \tau'_{xy} \\ \tau'_{yy} \\ u'\tau_{xy} + v'\tau_{yy} + u\tau'_{xy} + v\tau'_{yy} - q'_y \end{pmatrix}, \tag{16}$$

where $\bar{\bar{\tau}}'$ is the sensitivity of the viscous stress tensor and $\vec{q}'$ the sensitivity of the heat flux:

$$\begin{aligned} \tau'_{xx} &= \frac{2}{3}(\mu+\mu_t)\left(2\frac{\partial u'}{\partial x} - \frac{\partial v'}{\partial y}\right) + \frac{2}{3}(\mu'+\mu'_t)\left(2\frac{\partial u}{\partial x} - \frac{\partial v}{\partial y}\right), \\ \tau'_{xy} &= (\mu+\mu_t)\left(\frac{\partial u'}{\partial y} + \frac{\partial v'}{\partial x}\right) + (\mu'+\mu'_t)\left(\frac{\partial u}{\partial y} + \frac{\partial v}{\partial x}\right), \\ \tau'_{yy} &= \frac{2}{3}(\mu+\mu_t)\left(2\frac{\partial v'}{\partial y} - \frac{\partial u'}{\partial x}\right) + \frac{2}{3}(\mu'+\mu'_t)\left(2\frac{\partial v}{\partial y} - \frac{\partial u}{\partial x}\right), \end{aligned} \tag{17}$$

$$\begin{aligned} (q'_x, q'_y) = &-\lambda\left(\frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}\right)\left(\frac{\partial e'_i}{\partial x}, \frac{\partial e'_i}{\partial y}\right) \\ &-\lambda\left(\frac{\mu' Pr - \mu Pr'}{Pr^2} + \frac{\mu'_t Pr_t - \mu_t Pr'_t}{Pr_t^2}\right)\left(\frac{\partial e_i}{\partial x}, \frac{\partial e_i}{\partial y}\right). \end{aligned} \tag{18}$$

μ', Pr', μ'_t and Pr'_t are the sensitivities of corresponding coefficients. In the present work, we do not consider the sensitivity of turbulent quantities (see [21] for more details).

The boundary conditions for the sensitivity equations are obtained by differentiating the boundary conditions for the flow. In this perspective, we make the assumption that the geometry of the wall boundary may depend on a and is defined parametrically as:

$$\begin{aligned} I \subset \mathbb{R} &\to \mathbb{R}^2 \\ \xi &\mapsto \vec{x}_w(\xi, a). \end{aligned} \tag{19}$$

For the sake of simplicity, we assume that the geometry of the far field boundary does not depend on a. The wall boundary conditions for the sensitivity equations are obtained by applying the chain rule to equations (7). Note that, at the wall, the flow velocity is expressed as $\vec{U}(a, \vec{x}_w(a))$. Therefore, the boundary condition reads:

$$\frac{\partial \vec{U}}{\partial a}(\vec{x}_w) + \bar{\bar{\nabla}}\vec{U} \cdot \frac{\partial \vec{x}_w}{\partial a}(\vec{x}_w) = \left(\vec{U}' + \bar{\bar{\nabla}}\vec{U} \cdot \vec{x}'_w\right)(\vec{x}_w) = \vec{0}.$$

This yields a Dirichlet boundary condition for the sensitivity of the velocity vector:

$$\vec{U}'(\vec{x}_w) = -\left(\bar{\bar{\nabla}}\vec{U} \cdot \vec{x}'_w\right)(\vec{x}_w). \tag{20}$$

The wall boundary condition for the heat flux is obtained in a similar way:

$$\left((\vec{q}' + \bar{\bar{\nabla}}\vec{q} \cdot \vec{x}'_w) \cdot \vec{n} + \vec{q} \cdot \vec{n}'\right)(\vec{x}_w) = 0.$$

This yields the following boundary condition for the sensitivity of the heat flux at the wall:

$$(\vec{q}' \cdot \vec{n})\,(\vec{x}_w) = -\left((\bar{\bar{\nabla}}\vec{q} \cdot \vec{x}'_w) \cdot \vec{n} + \vec{q} \cdot \vec{n}'\right)(\vec{x}_w). \tag{21}$$

The boundary conditions at the far field boundary can be derived more easily, since we made the assumption that the geometry does not depend on a:

$$\left(\vec{\mathcal{F}}' \cdot \vec{n}\right)(\vec{x}_{far}) = \mathcal{F}'_0 \tag{22}$$

$$\left(\vec{\mathcal{G}}' \cdot \vec{n}\right)(\vec{x}_{far}) = \mathcal{G}'_0 \tag{23}$$

One underlines that, if the geometry of the boundaries depends on a, the boundary conditions for the sensitivity equations exhibit a dependency with respect to the flow solution derivatives. Therefore, the boundary condition is not known explicitly (see [5] for specific treatments).

3 Some properties of sensitivity equations

Property 3.1. *The sensitivity of the convective flux is linear in* $\mathbf{W}'$.

Proof. By definition, the sensitivity of the convective flux is:

$$\vec{\mathcal{F}}' = \frac{\partial}{\partial a}\vec{\mathcal{F}}(\mathbf{W}). \tag{24}$$

Thus, we have:

$$\vec{\mathcal{F}}' = \frac{\partial \vec{\mathcal{F}}(\mathbf{W})}{\partial \mathbf{W}} \cdot \frac{\partial \mathbf{W}}{\partial a} = \frac{\partial \vec{\mathcal{F}}(\mathbf{W})}{\partial \mathbf{W}} \cdot \mathbf{W}'. \tag{25}$$

□

Property 3.2. *The sensitivity of the convective flux has the same Jacobian matrix as the convective flux.*

Proof. According to the previous property, we have:

$$\frac{\partial \vec{\mathcal{F}}'}{\partial \mathbf{W}'} = \frac{\partial \vec{\mathcal{F}}(\mathbf{W})}{\partial \mathbf{W}} = \vec{\mathcal{A}}(\mathbf{W}). \tag{26}$$

where $\vec{\mathcal{A}}(\mathbf{W}) = (A_x(\mathbf{W}), A_y(\mathbf{W}))$ represents the Jacobian matrices of the convective flux. □

Property 3.3. *The convective part of the sensitivity equations is hyperbolic and is characterized by the same wave speeds as the convective part of the flow equations.*

Proof. According to (12), the convective part of the sensitivity equations reads in a non-conservative form:

$$\frac{\partial \mathbf{W}'}{\partial t} + \frac{\partial \vec{\mathcal{F}}'}{\partial \mathbf{W}'} \cdot \vec{\nabla}\mathbf{W}' = 0. \tag{27}$$

Accounting for the previous property:

$$\frac{\partial \mathbf{W}'}{\partial t} + \vec{\mathcal{A}}(\mathbf{W}) \cdot \vec{\nabla}\mathbf{W}' = 0 \tag{28}$$

This system is hyperbolic and has the same eigenvalues as the flow equations. □

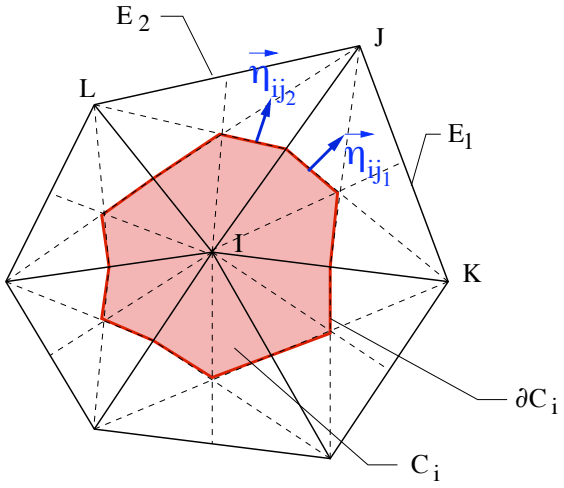


Figure 1: Control cell Ω_i around node i.

4 Numerical resolution

4.1 Mixed finite-volume / finite-element scheme

The flow equations (1) form a system of conservation laws, which is solved using a mixed finite-volume / finite-element formulation (see [4] for more details). Thus, we integrate (1) over a control volume Ω_i delimited by the surface Γ_i:

$$\int_{\Omega_i} \frac{\partial \mathbf{W}}{\partial t} + \nabla \cdot \vec{\mathcal{F}} \, d\Omega = \int_{\Omega_i} \nabla \cdot \vec{\mathcal{G}} \, d\Omega. \tag{29}$$

We note $\mathbf{W}_i$ the average value of $\mathbf{W}$ over the control volume Ω_i:

$$\int_{\Omega_i} \mathbf{W} \, d\Omega = \mathbf{W}_i \int_{\Omega_i} d\Omega = \mathbf{W}_i \mathcal{V}_i. \tag{30}$$

By using Green-Ostrogradski theorem, one introduces classically the convective flux through the interface Γ_i:

$$\mathcal{V}_i \frac{\partial \mathbf{W}_i}{\partial t} + \int_{\Gamma_i} \vec{\mathcal{F}} \cdot \vec{n} \, d\Gamma = \int_{\Omega_i} \nabla \cdot \vec{\mathcal{G}} \, d\Omega. \tag{31}$$

Now, we consider that Ω_i corresponds to a particular mesh cell, built around the mesh node i by gathering contributions from neighboring triangles T_k, $k \in \mathcal{T}(i)$ (vertex-centered approach), as illustrated in figure 1. The cell is surrounded by the neighboring cells Ω_j, $j \in \mathcal{N}(i)$. By decomposing the integrals, one may obtain:

$$\mathcal{V}_i \frac{\partial \mathbf{W}_i}{\partial t} + \sum_{j \in \mathcal{N}(i)} \int_{\Gamma_i \cap \Gamma_j} \vec{\mathcal{F}} \cdot \vec{n} \, d\Gamma = \sum_{k \in \mathcal{T}(i)} \int_{\Omega_i \cap T_k} \nabla \cdot \vec{\mathcal{G}} \, d\Omega. \tag{32}$$

The convective flux at the interface between the cells Ω_i and Ω_j, is then approximated using cell-averaged quantities $\mathbf{W}_i$ and $\mathbf{W}_j$:

$$\int_{\Gamma_i \cap \Gamma_j} \vec{\mathcal{F}} \cdot \vec{n} \, d\Gamma = \Phi(\mathbf{W}_i, \mathbf{W}_j, \vec{\eta}_{i,j}), \tag{33}$$

with:

$$\vec{\eta}_{i,j} = \int_{\Gamma_i \cap \Gamma_j} \vec{n} \, d\Gamma. \tag{34}$$

The diffusive flux is approximated using a finite-element discretization. It can be shown using integration by parts that (except at the boundary):

$$\int_{\Omega_i \cap T_k} \nabla \cdot \vec{\mathcal{G}} \, d\Omega = - \int_{T_k} \vec{\mathcal{G}} \cdot \nabla \vec{\varphi_i} \, d\Omega = \Psi(\mathbf{W}_k, \vec{\eta}_{i,T_k}), \tag{35}$$

where φ_i is the P1 basis function attached to the node i, $\mathbf{W}_k$ the quantities that prevail in each triangle T_k and:

$$\vec{\eta}_{i,T_k} = - \int_{T_k} \nabla \vec{\varphi_i} \, d\Omega. \tag{36}$$

Finally, one obtains the following semi-discretized formulation:

$$\mathcal{V}_i \frac{\partial \mathbf{W}_i}{\partial t} + \sum_{j \in \mathcal{N}(i)} \Phi(\mathbf{W}_i, \mathbf{W}_j, \vec{\eta}_{i,j}) = \sum_{k \in \mathcal{T}(i)} \Psi(\mathbf{W}_k, \vec{\eta}_{i,T_k}) \tag{37}$$

Since the sensitivity equations are formally similar to the flow equations, the same finite-volume approach is applied to equations (12):

$$\int_{\Omega_i} \frac{\partial \mathbf{W}'}{\partial t} + \nabla \cdot \vec{\mathcal{F}}' \, d\Omega = \int_{\Omega_i} \nabla \cdot \vec{\mathcal{G}}' \, d\Omega. \tag{38}$$

By introducing the cell-averaged quantity $\mathbf{W}'_i$, one can obtain:

$$\mathcal{V}_i \frac{\partial \mathbf{W}'_i}{\partial t} + \sum_{j \in \mathcal{N}(i)} \int_{\Gamma_i \cap \Gamma_j} \vec{\mathcal{F}}' \cdot \vec{n} \, d\Gamma = \sum_{k \in \mathcal{T}(i)} \int_{\Omega_i \cap T_k} \nabla \cdot \vec{\mathcal{G}}' \, d\Gamma. \tag{39}$$

With the assumption that approximate fluxes are defined for the sensitivity equations, in the spirit of those for the flow equations, one obtain a semi-discretized formulation similar to (37):

$$\mathcal{V}_i \frac{\partial \mathbf{W}'_i}{\partial t} + \sum_{j \in \mathcal{N}(i)} \Theta(\mathbf{W}'_i, \mathbf{W}'_j, \vec{\eta}_{i,j}) = \sum_{k \in \mathcal{T}(i)} \Upsilon(\mathbf{W}'_k, \vec{\eta}_{i,T_k}). \tag{40}$$

The fact that semi-discretized flow equations (37) and sensitivity equations (40) are formally similar allows to use the same numerical methods to solve them. In particular, the spatial scheme could be identical, provided that the approximate convective and diffusive terms for the sensitivity equations are defined consistently.

4.2 Convective flux

Any standard approximate Riemann solver could be used, for the flow as well as sensitivity fluxes. In the present study, the convective flux for the flow Φ is computed using the HLLC method [1, 20]. Concerning the sensitivity flux, the HLL method is presently preferred for its ease of implementation, as suggested in [10], and described below.

At each interface between two cells, characterized by two sensitivity states $\mathbf{W}'_i$ and $\mathbf{W}'_j$, the HLL Riemann solver is based on a three-state approach: an intermediate region is defined, characterized by the uniform state $\mathbf{W}'_\star$, limited by two waves propagating respectively at the

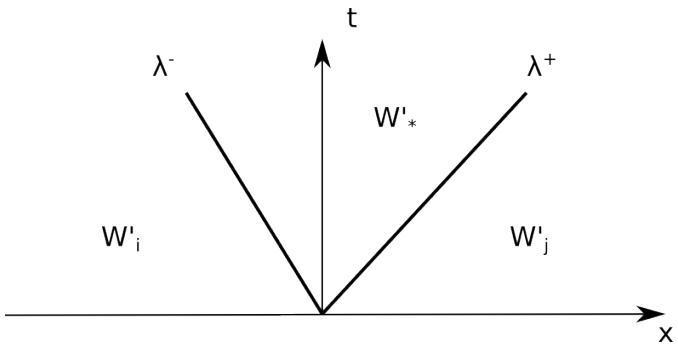


Figure 2: Configuration adopted for HLL flux.

speed λ^- and λ^+, which are considered as discontinuities. This configuration is illustrated by Fig. (2), for the one-dimensional subsonic case $\lambda^- < 0 < \lambda^+$:

At the two discontinuities the following jump relationships hold for the normal flux $F' = \vec{\mathcal{F}}' \cdot \vec{n}$, according to Rankine-Hugoniot conditions:

$$\begin{aligned} F'_\star - F'_i &= (\mathbf{W}'_\star - \mathbf{W}'_i)\, \lambda^-, \\ F'_j - F'_\star &= (\mathbf{W}'_j - \mathbf{W}'_\star)\, \lambda^+. \end{aligned} \tag{41}$$

The resolution of this system yields the following expression for the convective sensitivity flux:

$$\Theta_{HLL} = F'_\star = \frac{\lambda^+ F'_i - \lambda^- F'_j}{\lambda^+ - \lambda^-} - \frac{\lambda^+ \lambda^-}{\lambda^+ - \lambda^-} (\mathbf{W}'_i - \mathbf{W}'_j). \tag{42}$$

An estimate of the wave speeds λ^- and λ^+ is still required to define the flux. As explained above, the wave speeds associated to the sensitivity equations are the same as those for the flow equations. Therefore, the following wave speeds are chosen [1, 20], yielding a flux expression Eq. (42) adapted to all flow regimes:

$$\begin{aligned} \lambda^+ &= \mathrm{Max}(U_i^n + c_i, U_j^n + c_j, 0), \\ \lambda^- &= \mathrm{Min}(U_i^n - c_i, U_j^n - c_j, 0), \end{aligned} \tag{43}$$

where $U^n = \overrightarrow{U} \cdot \vec{\eta}_{i,j}$ denotes the velocity normal to the cell interface and c the sound speed.

A high-order flux reconstruction is finally obtained for the flow as well as sensitivities, by using extrapolated state variables $\widetilde{\mathbf{U}}'$ to compute the terms in Eq. (42), in the spirit of the MUSCL technique from Van Leer [20], with:

$$\begin{aligned} \widetilde{\mathbf{U}}'_i &= \mathbf{U}'_i + \frac{1}{2} (\overrightarrow{\nabla} \mathbf{U}'_i) \cdot \overrightarrow{IJ}, \\ \widetilde{\mathbf{U}}'_j &= \mathbf{U}'_j + \frac{1}{2} (\overrightarrow{\nabla} \mathbf{U}'_j) \cdot \overrightarrow{JI}. \end{aligned} \tag{44}$$

Note that extrapolation is carried out for primitive variables $\mathbf{U}'$, and not for conservative variables $\mathbf{W}'$. We do not introduce slope limiters here, due to the fact that only subsonic flows are considered in this study.

We underline that, using such an approach, the evaluation of the convective flux for sensitivities is mostly based on methods already used for the flow analysis. Thus, only a few specific implementations are necessary.

4.3 Diffusive term

Due to the fact that the gradient fields are constant over each triangle T_k, the diffusive term for the flow is approximated by the following expression:

$$\Psi(\mathbf{W}_k, \vec{\eta}_{i,T_k}) = \vec{\mathcal{G}}(\mathbf{W}_k) \cdot \vec{\eta}_{i,T_k}, \tag{45}$$

the later term being only geometrical. The same expression holds for the sensitivity diffusive flux:

$$\Upsilon(\mathbf{W}'_k, \vec{\eta}_{i,T_k}) = \vec{\mathcal{G}}'(\mathbf{W}'_k) \cdot \vec{\eta}_{i,T_k} \tag{46}$$

4.4 Temporal scheme

A three-step backward finite-difference scheme is used to obtain a second-order accurate approximation of the unsteady term for the flow equations:

$$\mathcal{V}_i \frac{\partial \mathbf{W}_i}{\partial t} \approx \frac{\mathcal{V}_i}{2\Delta t} \left(3\mathbf{W}_i^{n+1} - 4\mathbf{W}_i^{n} + \mathbf{W}_i^{n-1}\right). \tag{47}$$

Additionally, a local pseudo time-step $\Delta\tau_i$ is added to increase the diagonal dominance of the resulting system. In this dual time-stepping procedure, the solution at any time step $n+1$ is obtained as the limit of an internal iterative process indexed by k:

$$\mathcal{V}_i \frac{\partial \mathbf{W}_i}{\partial t} \approx \frac{\mathcal{V}_i}{2\Delta t} \left(3\mathbf{W}_i^{(n+1,k+1)} - 4\mathbf{W}_i^{n} + \mathbf{W}_i^{n-1}\right) + \frac{\mathcal{V}_i}{\Delta\tau_i} \left(\mathbf{W}_i^{(n+1,k+1)} - \mathbf{W}_i^{(n+1,k)}\right). \tag{48}$$

In terms of variations, the later equation reads:

$$\mathcal{V}_i \frac{\partial \mathbf{W}_i}{\partial t} \approx \left(\frac{3\mathcal{V}_i}{2\Delta t} + \frac{\mathcal{V}_i}{\Delta\tau_i}\right) \delta\mathbf{W}_i + \frac{3\mathcal{V}_i}{2\Delta t} \delta\mathbf{W}_i^{n} - \frac{\mathcal{V}_i}{\Delta t} \delta\mathbf{W}_i^{n-1}, \tag{49}$$

with:

$$\delta\mathbf{W}_i = \mathbf{W}_i^{(n+1,k+1)} - \mathbf{W}_i^{(n+1,k)} \quad \delta\mathbf{W}_i^{n} = \mathbf{W}_i^{(n+1,k)} - \mathbf{W}_i^{n} \quad \delta\mathbf{W}_i^{n-1} = \mathbf{W}_i^{n} - \mathbf{W}_i^{n-1} \tag{50}$$

The first term in Eq.(49) is the state increment to be computed at each step k, whereas other terms are known and treated as source terms. The same scheme is adopted for the sensitivity variables as well.

4.5 Iterative procedure

The temporal scheme with the implicit discretization of convective and diffusive terms yield the following non-linear problem, to be solved at each step:

$$\mathbf{D}(\mathbf{W}^k)\, \delta\mathbf{W} + \mathbf{R}(\mathbf{W}^{k+1}) = \mathbf{S}(\mathbf{W}^k, \mathbf{W}^n, \mathbf{W}^{n-1}). \tag{51}$$

The vector $\mathbf{R}$ results from the assembly of the high-order convective and diffusive terms for each cell, the vector $\mathbf{S}$ corresponds to the unsteady source terms and $\mathbf{D}$ is a diagonal matrix corresponding to the pseudo and physical unsteady terms:

$$\mathbf{D}_{ii} = \mathcal{V}_i \left(\frac{3}{2\Delta t} + \frac{1}{\Delta\tau_i(\mathbf{W}^k)}\right), \tag{52}$$

where $\mathcal{V}_i$ is the volume of the cell i, Δt the physical time-step and $\Delta\tau_i$ a local pseudo time-step adjusted according to stability criteria.

This non-linear problem for the flow is solved using an implicit approach based on an approximate Jacobian matrix [4]. More precisely, a conservative state increment $\delta\mathbf{W}$ is computed at each iteration k according to a Newton-like method based on the linearization of the residual vector $\mathbf{R}$, yielding the following linear system:

$$\left(\mathbf{D}(\mathbf{W}^k) + \mathbf{J}(\mathbf{W}^k)\right)\delta\mathbf{W} = -\mathbf{R}(\mathbf{W}^k) + \mathbf{S}(\mathbf{W}^k, \mathbf{W}^n, \mathbf{W}^{n-1}). \tag{53}$$

$\mathbf{J}$ is a sparse matrix resulting from an approximate linearization of the right-hand side vector. It is based on the exact linearization of the diffusive terms, but on the linearization of the first-order convective flux proposed by Rusanov, yielding a more robust and efficient procedure [14]. Finally, the linear system is inverted using a standard SGS (Symmetric Gauss-Seidel) procedure. Due to the approximation of the Jacobian, a full inversion is not required and a reasonable reduction of the linear residuals is only carried out in practice. This iterative procedure is used until convergence of non-linear residuals for each time step.

Although the sensitivity equations are linear, a similar iterative procedure is used, for two reasons: first, the corresponding linear system can be ill-conditioned, yielding a computational time for its inversion similar to a full non-linear flow resolution. Second, we pay attention to limit as much as possible new implementations, specific to the sensitivity equations, and promote re-use of numerical methods already implemented for the flow analysis.

As explained in a previous section, the convective flux for the sensitivity variables $\vec{\mathcal{F}}'(\mathbf{W}')$ has the same Jacobian matrix as the flux for the flow variables:

$$\frac{\partial\vec{\mathcal{F}}'}{\partial\mathbf{W}'} = \frac{\partial\vec{\mathcal{F}}}{\partial\mathbf{W}} = \vec{\mathcal{A}}(\mathbf{W}). \tag{54}$$

This property is also true for the diffusive terms. Therefore, the Jacobian matrix $\mathbf{J}(\mathbf{W}_k)$ that appears in the iterative procedure for the flow Eq. (55) can also be employed in a similar procedure for the sensitivities. Moreover, the wave speeds related to the sensitivity variables are also the same as those related to the flow variables. As consequence, the unsteady term $\mathbf{D}(\mathbf{W}_k)$ can be maintained as well. Finally, one can observe that an iterative procedure for the sensitivity variables can be defined, based on the same implicit part as the one used for the flow:

$$\left(\mathbf{D}(\mathbf{W}^k) + \mathbf{J}(\mathbf{W}^k)\right)\delta\mathbf{W}' = -\mathbf{R}'(\mathbf{W}'^k, \mathbf{W}^k) + \mathbf{S}(\mathbf{W}'^k, \mathbf{W}'^n, \mathbf{W}'^{n-1}) \tag{55}$$

where $\delta\mathbf{W}'$ denotes the increment of the sensitivity variables at iteration k and the vector $\mathbf{R}'$ results from the assembly of the high-order convective and diffusive terms for the sensitivities. We underline that this residual term is actually the only one to be implemented for sensitivity analysis in an existing flow solver.

5 Implementation

The numerical methods described above are implemented in the Num3sis platform (`http://num3sis.inria.fr`) developed at Inria (C/C++ language), in a versatile way, without any assumption on the choice of the sensitivity parameter a. The geometric implementation is generic (1D, 2D or 3D). A domain partitioning approach is used for parallel computations.

6 Verification of sensitivity analysis

To verify the implementation of the sensitivity equations, we compute a neighboring solution according to a first-order extrapolation $\mathbf{U}(a)+\mathbf{U}'\delta a$ and compare it with the solution $\mathbf{U}(a+\delta a)$. The difference should be of order $\mathcal{O}(\delta^2 a)$. Therefore, we plot the L2 norm of this difference w. r. t. δa in log-log scale.

6.1 Steady inviscid flow in a nozzle

We consider as first test-case the steady inviscid subsonic flow in a two-dimensional nozzle. The Mach number is set to the value $M = 0.15$. Note that the transonic case yields specific difficulties due to the presence of the shock waves, that are not treated presently (see [10] for instance). Fig. (3) shows a description of the test-case and Fig. (4) the grid used (23 489 nodes). Note that the mesh should be fine enough to make the approximation error negligible. The local time-step is chosen so that the local CFL number is 10 000. Computations are carried out using four processors. The iterative convergence is plotted in Fig. (5) and the flow field is illustrated in Fig. (6).

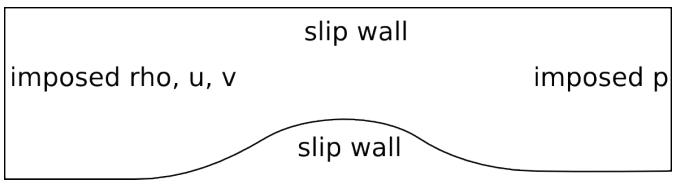


Figure 3: Nozzle test-case configuration

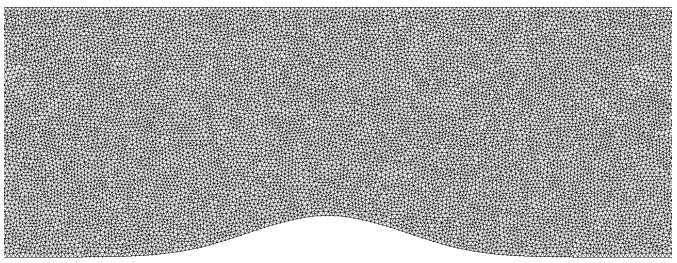


Figure 4: Nozzle case: mesh

Regarding the sensitivity analysis, three different parameters are considered here: the inlet velocity u_∞ and density ρ_∞, and the outlet pressure p_∞. The iterative convergence of the sensitivity residuals is depicted in Fig. (5). Note that the residuals are computed here before the linear system is solved, as for the flow. One can see that the sensitivities have globally the same convergence rate as the flow. This is due to the fact that sensitivities depend on the flow, therefore a better convergence cannot be expected. Obviously, one could solve the (linear) sensitivity problem only at flow convergence, to reduce the computational cost in steady cases.

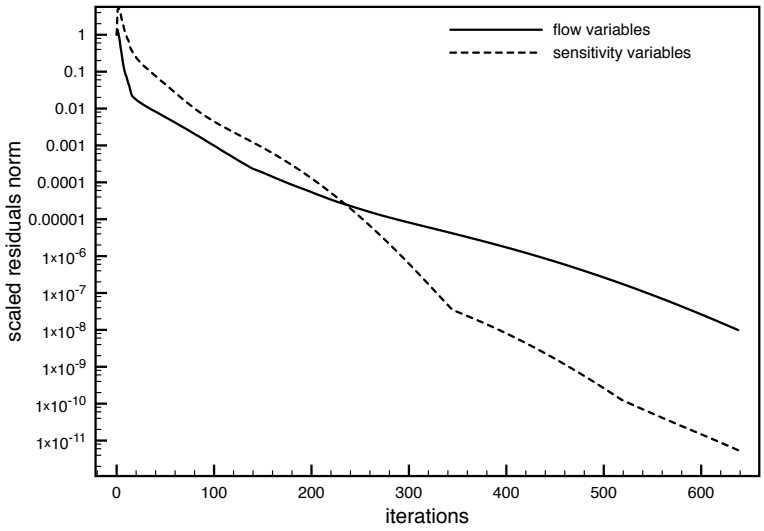


Figure 5: Nozzle case: iterative convergence for the flow and sensitivity w.r.t. inlet velocity u_∞

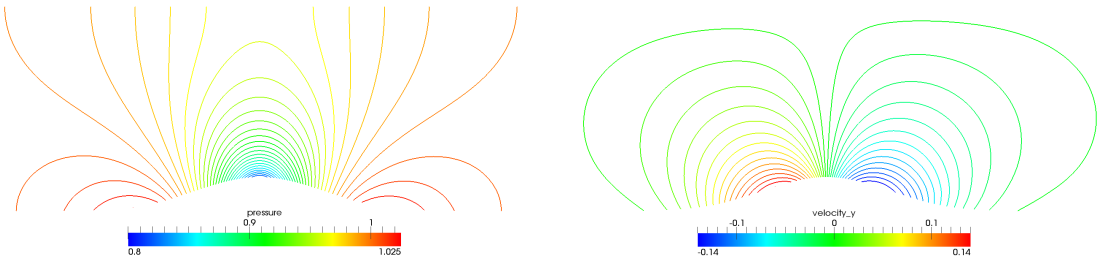


Figure 6: Nozzle case: iso-p (left) and iso-v (right) contours.

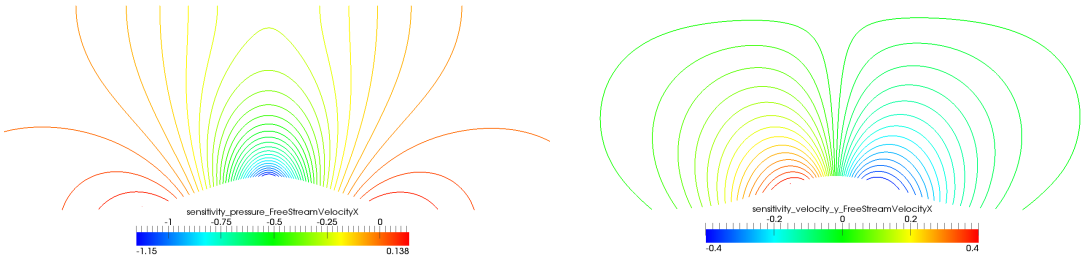


Figure 7: Nozzle case: iso-p' (left) and iso-v' (right) contours w.r.t. inlet velocity u_∞.

The sensitivity fields w.r.t. u_∞ are illustrated in Fig. (7). As can be observed, the sensitivity fields are very close to the flow fields, for this rather simple problem.

Finally, the results of the verification exercises, in terms of error of linear extrapolation, are plotted in Figs (8-10). As can be seen, a good agreement with the theoretical results are obtained, except of the sensitivity of the velocity w.r.t. the inlet density. This can be explained by the fact that, for this case, the velocity does not depends so much on the inlet density. Therefore, the sensitivity magnitude is very low. For small perturbations, the extrapolation error becomes of the same order as the approximation error, yielding a gap with the reference slope. In fact, this test underlines the need for an automated grid refinement procedure.

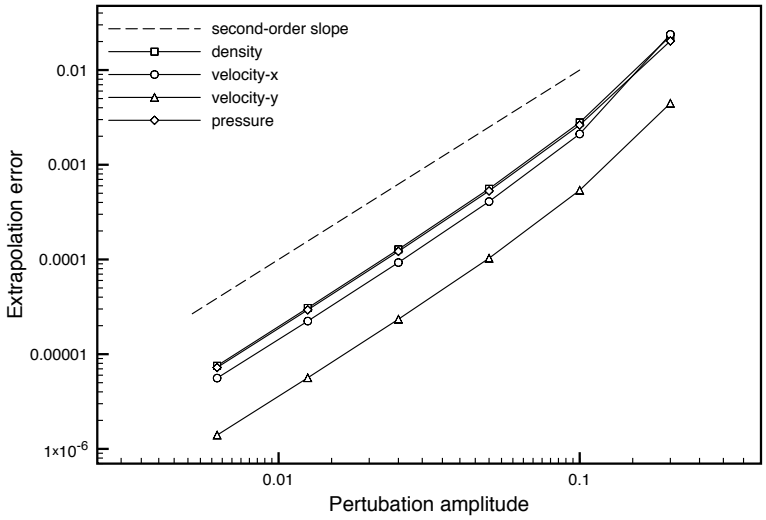


Figure 8: Nozzle case: error of linear extrapolation of flow fields w.r.t. inlet velocity u_∞

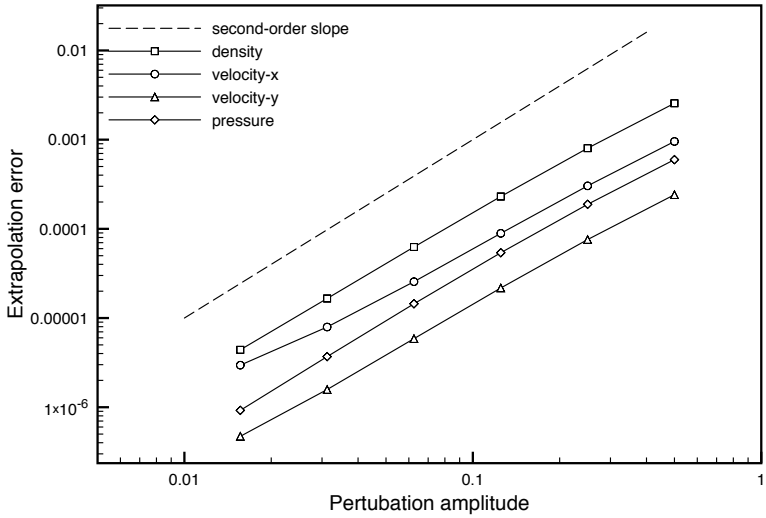


Figure 9: Nozzle case: error of linear extrapolation of flow fields w.r.t. outlet pressure p_∞

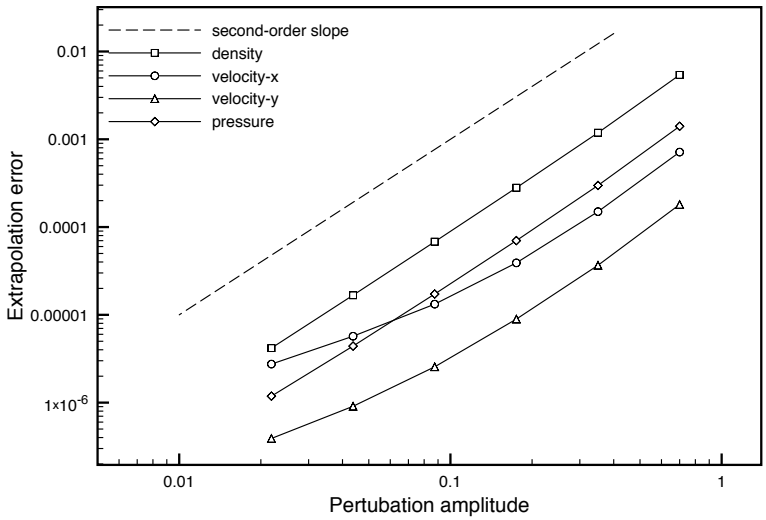


Figure 10: Nozzle case: error of linear extrapolation of flow fields w.r.t. inlet density ρ_∞

6.2 Steady viscous flow around a cylinder

As second test-case, we consider the classical problem of the flow around a cylinder, for a Reynolds number $Re = 40$ and Mach number $M = 0.15$, for which a steady flow including two symmetric recirculation areas is expected. Free-stream boundary conditions are prescribed as for the nozzle case, whereas a no-slip condition is imposed on the cylinder wall. The grid counts 99791 nodes (see Fig. (11)). The local time-step is chosen so that the local CFL number is 1000. Computations are carried out using four processors. The iterative convergence is plotted in Fig. (12) and the streamlines are depicted in Fig. (13).

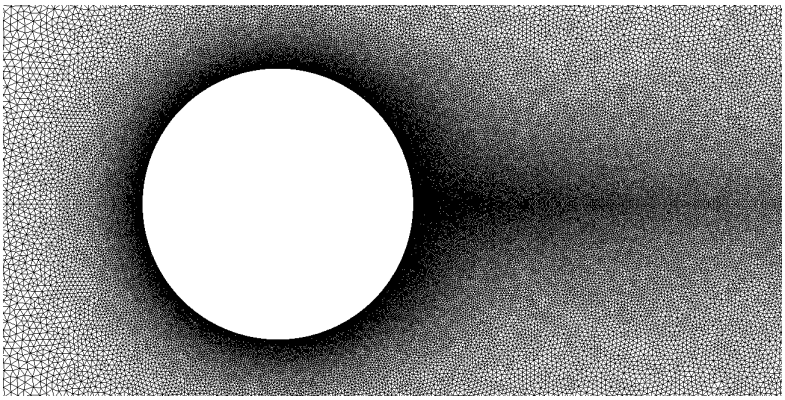


Figure 11: Cylinder case: mesh

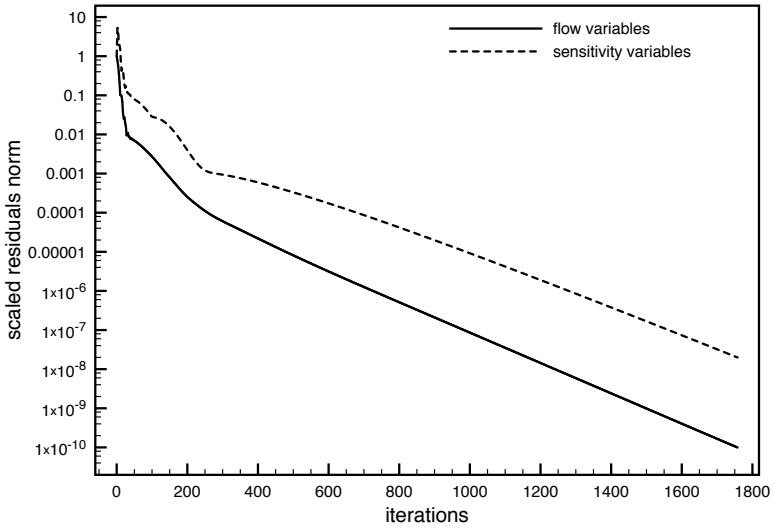


Figure 12: Cylinder case: iterative convergence of the flow and sensitivity w.r.t. viscosity μ

We choose as sensitivity parameters the free-stream velocity u_∞ and the viscosity coefficient μ. The convergence of the sensitivity residuals (parameter μ) exhibits the same convergence rate

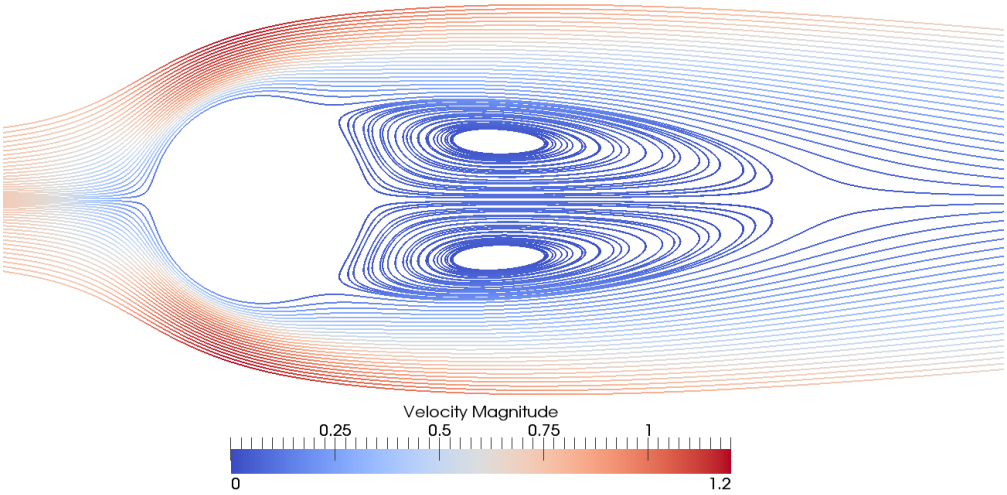


Figure 13: Cylinder case: streamlines colored by velocity magnitude

as the flow, as can be seen in Fig. (12). As for the previous case, we compute the L2 norm of the error between the linearly extrapolated state and the non-linear perturbed state, for different perturbation amplitudes. Results can be seen in Fig. (14) and Fig. (16), for the two parameters. The same exercise is also carried out for the drag, as shown in Fig (15) and Fig. (17). A good agreement is obtained with the theory, even if approximation error is present for the smallest perturbations again.

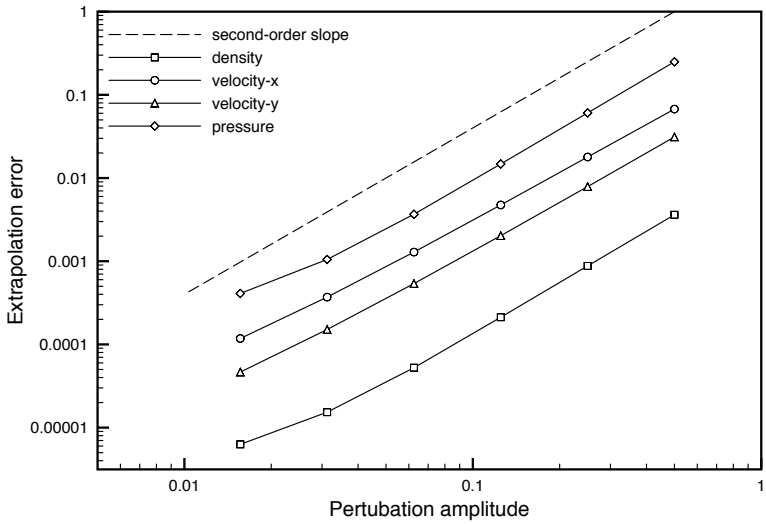


Figure 14: Cylinder case: error of linear extrapolation of flow fields w.r.t. free-stream velocity u_∞

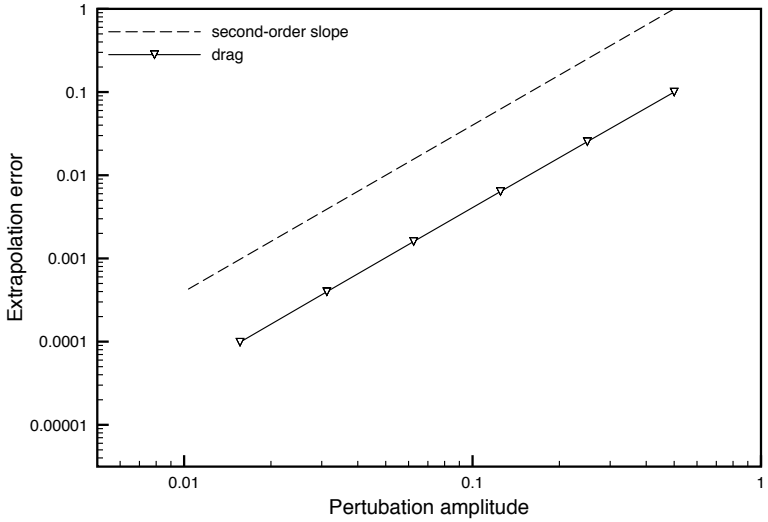


Figure 15: Cylinder case: error of linear extrapolation of drag w.r.t. free-stream velocity u_∞

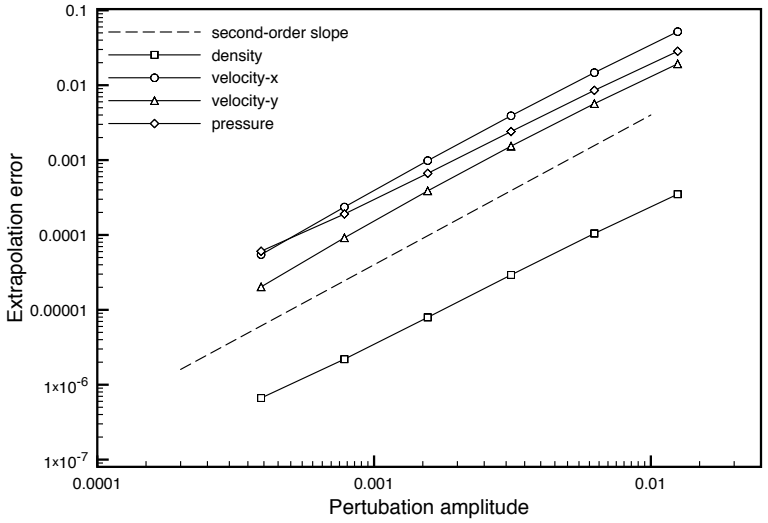


Figure 16: Cylinder case: error of linear extrapolation of flow fields w.r.t. viscosity μ

Finally, we demonstrate some fields extrapolation results in Figs. (18-20). Some iso-contours are plotted for the field corresponding to a reference parameter value, for the linearly extrapolated field and for the field corresponding to the perturbed parameter value. A very good agreement is obtained again.

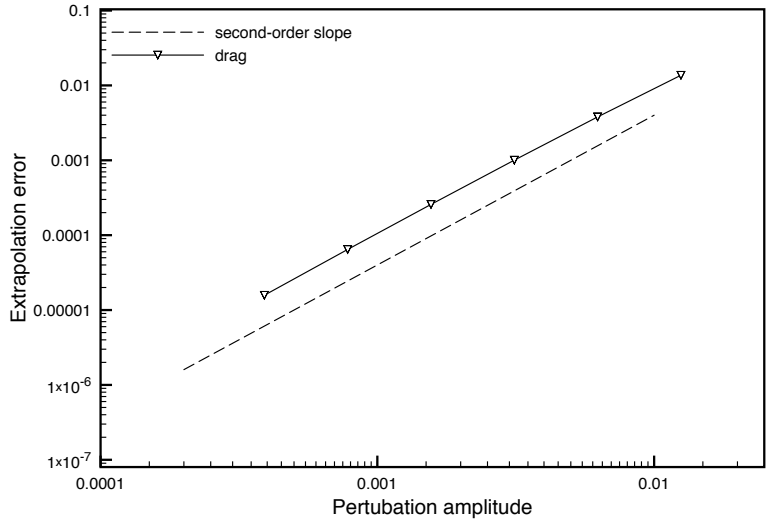


Figure 17: Cylinder case: error of linear extrapolation of drag w.r.t. viscosity μ

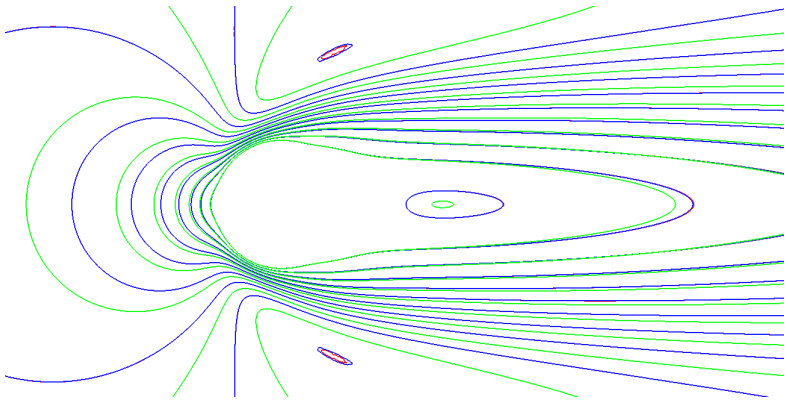


Figure 18: Cylinder case: linear extrapolation of velocity field u w.r.t. free-stream velocity u_∞ (reference state $u(u_\infty)$ in green, extrapolated state $u(u_\infty) + u'\delta u_\infty$ in red, non-linear perturbed state $u(u_\infty + \delta u_\infty)$ in blue, for $\delta u_\infty = u_\infty/16$).

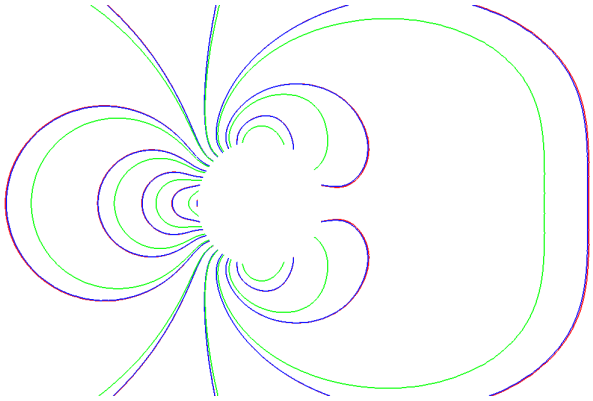


Figure 19: Cylinder case: linear extrapolation of pressure field p w.r.t. free-stream velocity u_∞ (reference state $p(u_\infty)$ in green, extrapolated state $p(u_\infty) + p'\delta u_\infty$ in red, non-linear perturbed state $p(u_\infty + \delta u_\infty)$ in blue, for $\delta u_\infty = u_\infty/16$).

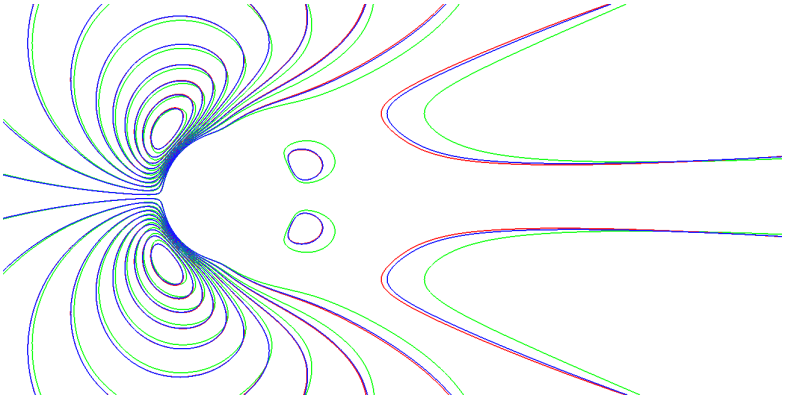


Figure 20: Cylinder case: linear extrapolation of velocity field v w.r.t. viscosity μ (reference state $v(\mu)$ in green, extrapolated state $v(\mu) + v'\delta\mu$ in red, non-linear perturbed state $v(\mu + \delta\mu)$ in blue, for $\delta\mu = \mu/4$).

6.3 Boundary layer flow with periodic actuation

We consider now an unsteady viscous flow including a periodic suction / blowing device, in the spirit of synthetic jet actuators. This device, of length $h/10$, is located at a distance h from the stagnation point, in a classical flat plate boundary layer flow, as illustrated in Fig. (21). The distance between the plate and the inlet is $2h$, and the plate length $6h$. The channel width is $4h$. The Mach number is $M = 0.15$ and the Reynolds number based on the length h is $Re = 100$. The grid counts 64235 nodes, including refinement areas at the stagnation point and around the jet, as shown in Fig. (22). The computations are carried out using 6 processors. The non-dimensional jet frequency is $F_{jet} = 10$. More precisely, the jet boundary condition is implemented as a Dirichlet condition for the velocity:

$$v(x,t) = V_{jet}\ Sin(2\pi F_{jet} t)\ d(x), \tag{56}$$

where $d(x)$ corresponds to a square sine distribution. $V_{jet} = u_\infty$ is the jet amplitude and will be considered as sensitivity parameter. Therefore, the boundary condition for the sensitivity is:

$$v'(x,t) = Sin(2\pi F_{jet} t)\ d(x), \tag{57}$$

and zero at other boundaries. The flow is initialized with uniform fields and zero fields for the sensitivity problem. The physical time is $\Delta t = 5.\,10^{-4}$, which corresponds to 200 time steps per actuation period.

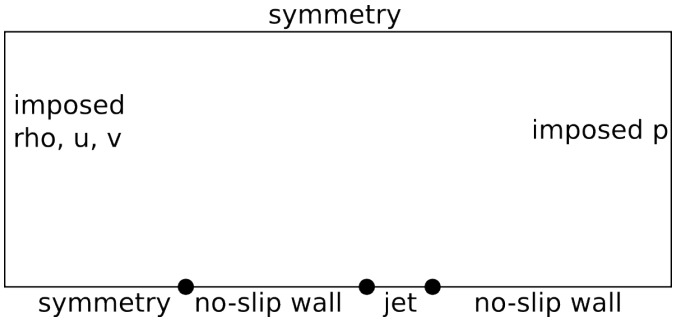


Figure 21: Jet test-case configuration

During the simulation time, a boundary layer is developing along the plate, while the jet is generating flow oscillations. The velocity vectors in the vicinity of the jet slot, for different excitation phases, are represented on Fig. (23). The oscillations of the velocity component u can be seen more globally in Fig. (24), with the time-dependent sensitivity field u'. Flow and sensitivity do not exhibit the same pattern here.

As for the cylinder case, we compare the flow fields predicted by linear extrapolation $U(V_{jet}) + U'\delta V_{jet}$ with the non-linear flow for perturbed condition $U(V_{jet} + \delta V_{jet})$. u and p variables can be seen in Fig. (25-26), for a perturbation $\delta V_{jet} = V_{jet}/4$. As observed, a very satisfactory agreement is found, in spite of the flow complexity. Similarly, the drag history, for the extrapolated case and non-linear perturbed case, can be seen in Fig. (27). Again, sensitivity computation allows a very accurate prediction of the neighboring drag value and its dynamics.

Regarding the computational efficiency of the procedure, the ratio of the computational time for the simulation plus sensitivity analysis over the computational time for a simulation alone,

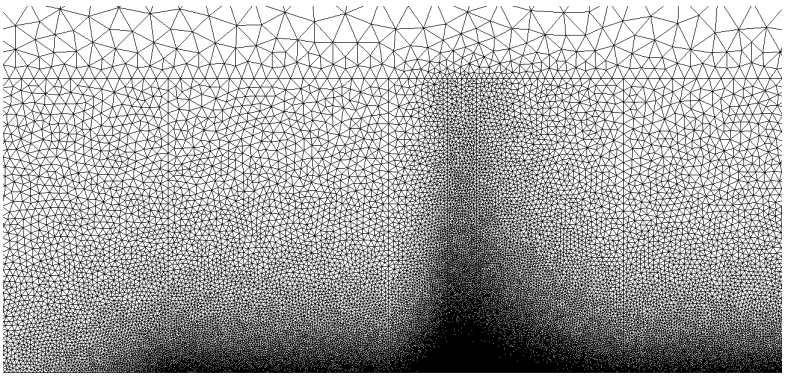


Figure 22: Jet case: mesh

for this unsteady case, has been found equal to:

$$r = \frac{\text{CPU}(U + U')}{\text{CPU}(U)} = 1.42 \tag{58}$$

The extra cost for the sensitivity analysis is due to the sensitivity residuals computation and system solving. This seems to be reasonable. Moreover, we underline that parallel computing could be used for the resolution of the sensitivity equations, when different parameters are considered, because sensitivity equations are independent from each other.

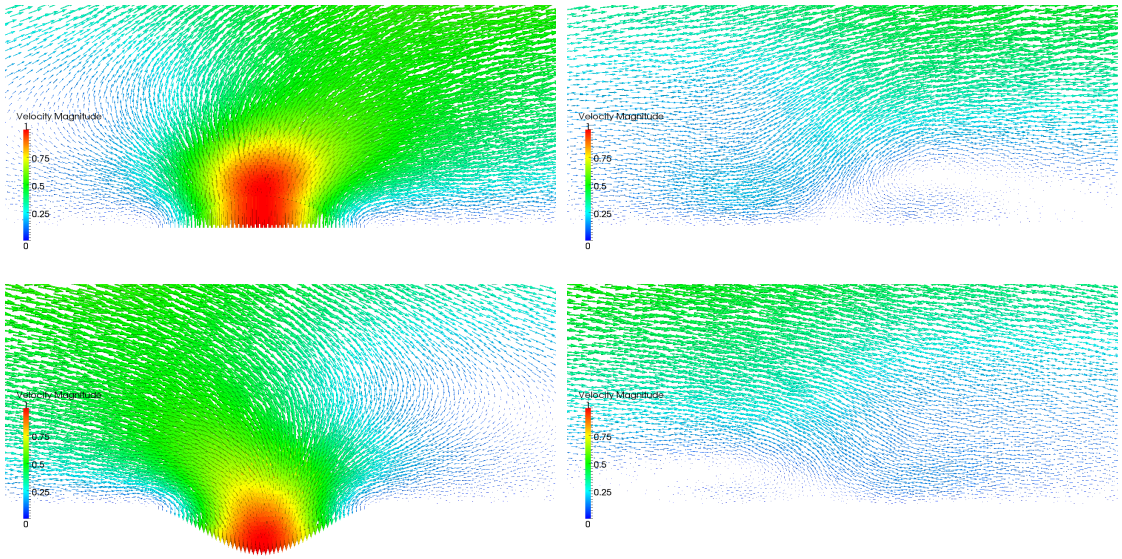


Figure 23: Jet case: velocity vectors at different phases (blowing, between blowing and suction, suction, between suction and blowing).

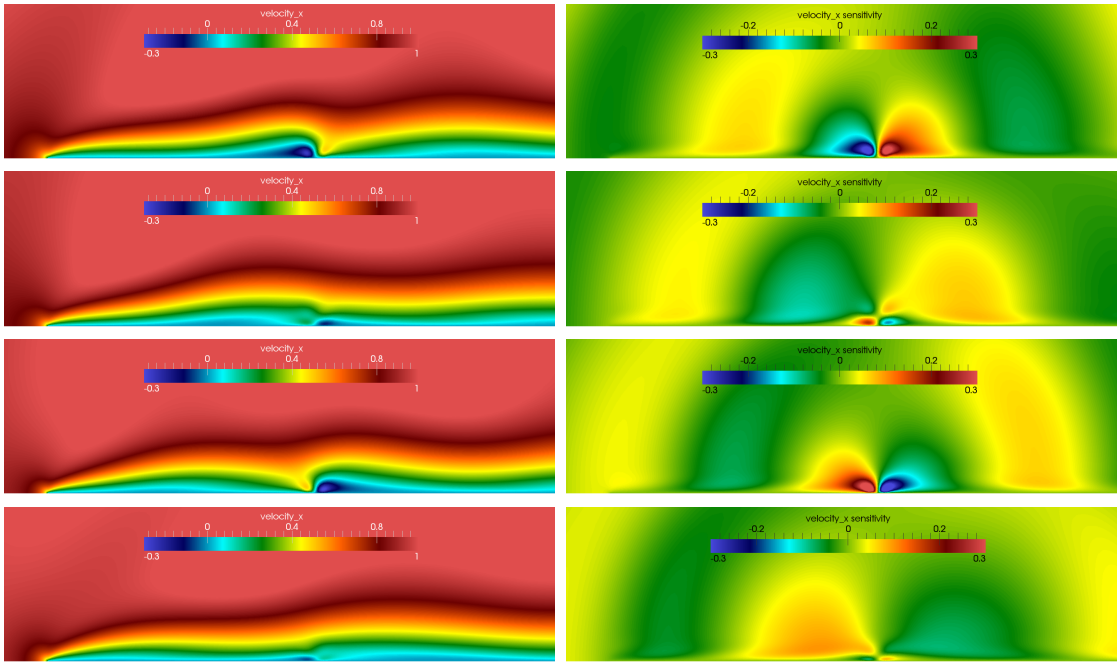


Figure 24: Jet case: u (left) and u' (right) fields at different phases (blowing, between blowing and suction, suction, between suction and blowing).

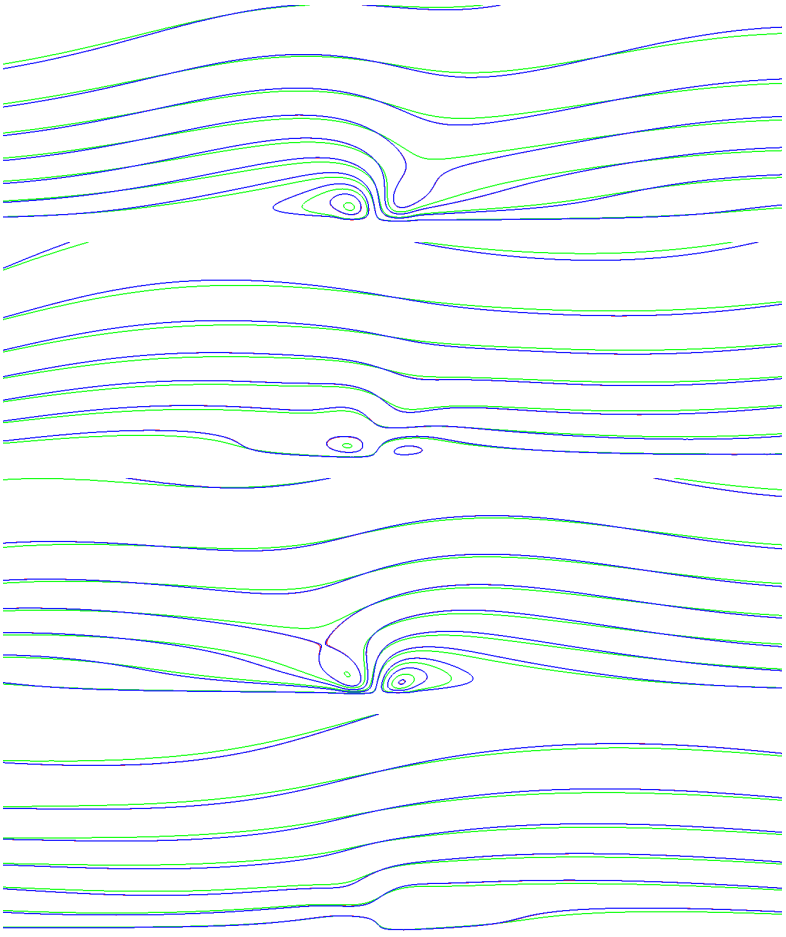


Figure 25: Jet case: linear extrapolation of velocity field u w.r.t. jet amplitude V_{jet} for different phases (reference state $u(V_{jet})$ in green, extrapolated state $u(V_{jet}) + u'\delta V_{jet}$ in red, non-linear perturbed state $u(V_{jet} + \delta V_{jet})$ in blue, for $\delta V_{jet} = V_{jet}/4$).

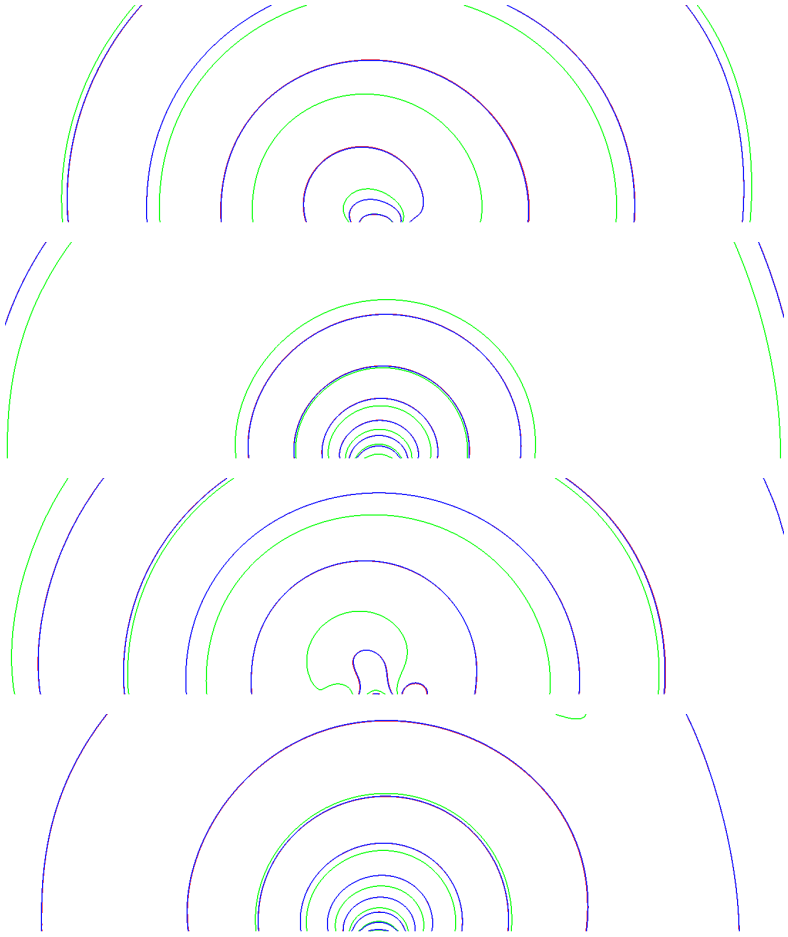


Figure 26: Jet case: linear extrapolation of pressure field p w.r.t. jet amplitude V_{jet} for different phases (reference state $p(V_{jet})$ in green, extrapolated state $p(V_{jet}) + p'\delta V_{jet}$ in red, non-linear perturbed state $p(V_{jet} + \delta V_{jet})$ in blue, for $\delta V_{jet} = V_{jet}/4$).

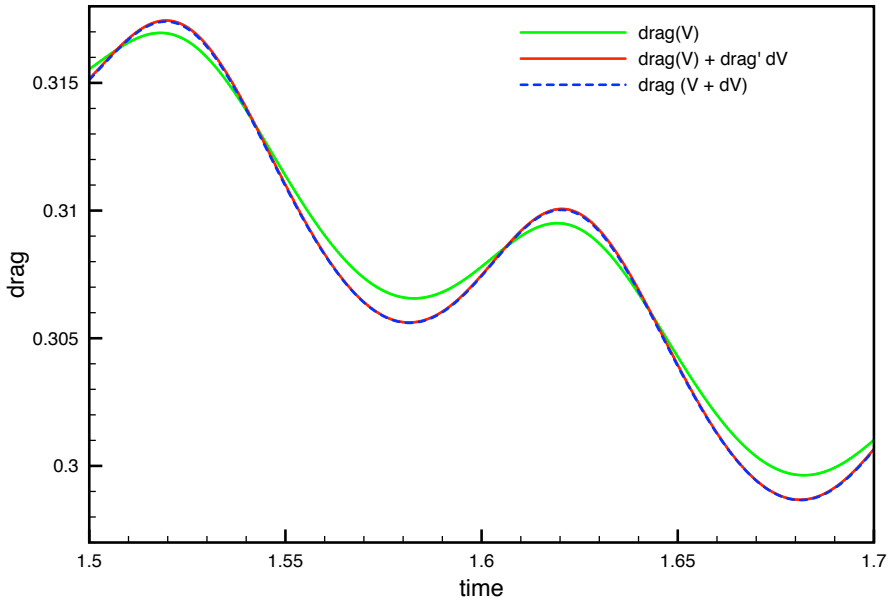


Figure 27: Jet case: extrapolation of time-dependent drag w.r.t. jet amplitude (reference state $F_X(V_{jet})$ in green, extrapolated state $F_X(V_{jet}) + F'_X \delta V_{jet}$ in red, non-linear perturbed state $F_X(V_{jet} + \delta V_{jet})$ in blue, for $\delta V_{jet} = V_{jet}/4$).

6.4 3D unsteady flow

The previous test-case is extended to a three-dimensional one by extrusion of the plate along z axis. The computational domain width is h, limited by two symmetry planes. The grid counts 479 319 cells and is coarser than the one used for the two-dimensional test. The jet is designed to be squared. The velocity distribution at the jet boundary condition is obtained by a tensor product:

$$v(x,z,t) = V_{jet}\ Sin(2\pi F_{jet} t)\ d(x)\ d(z). \tag{59}$$

Except for the geometry, this test-case is identical to the previous one and is intended to verify the three dimensional characteristics of the implemented methods. The computations are carried out using 32 processors. The flow is illustrated in Fig. (28) by the streamwise velocity field u. As previously, to verify the sensitivity analysis, we consider some comparisons between linearly extrapolated fields and neighboring non-linear flow solutions.

Thus, two components of the velocity fields are compared for two sections in Figs. (29 - 30), with a satisfactory agreement although a corser grid is used. Regarding the drag, we compare the drag variation $F_X(V_{jet}+\delta V_{jet}) - F_X(V_{jet})$ with the linear variation based on the sensitivity $F'_X \delta V_{jet}$ in Fig. (31). As seen, the drag variation is very small, due to the domain extrusion. However the linear and non-linear drag variations are similar.

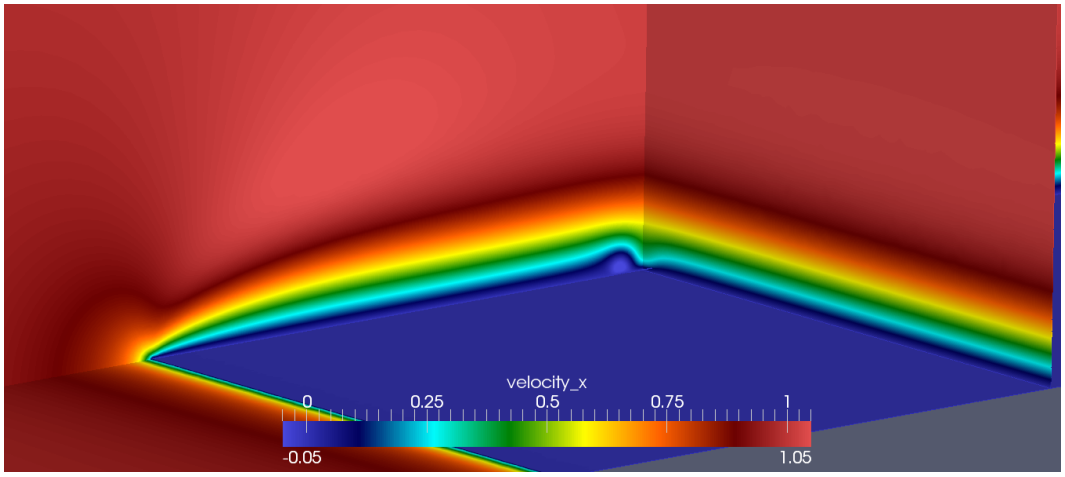


Figure 28: 3D jet case: streamwise velocity u field in the planes $y = 0$ (plate), $z = 0$ (symmetry) and $x = 1.04h$ (section just downstream the jet), during blowing.

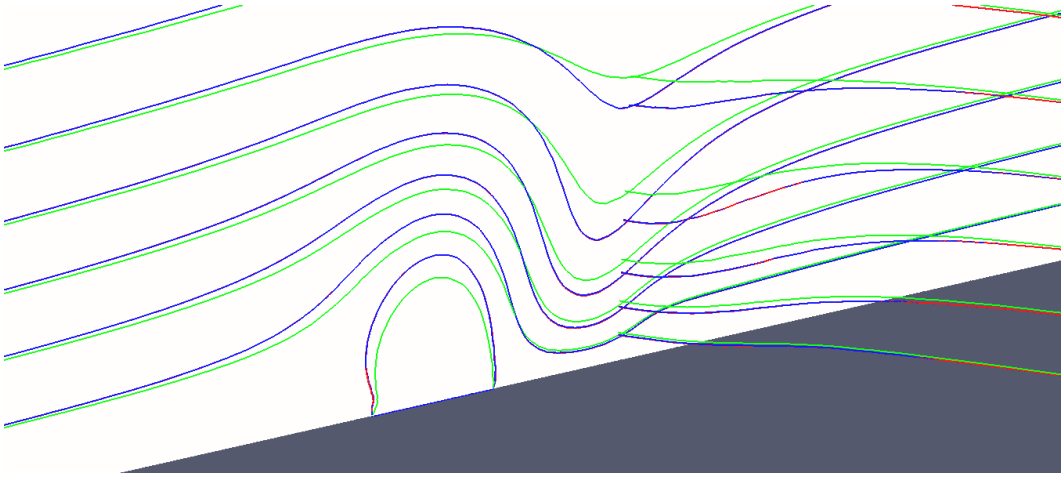


Figure 29: 3D jet case: linear extrapolation of streamwise velocity field u w.r.t. jet amplitude V_{jet} during blowing (reference state $u(V_{jet})$ in green, extrapolated state $u(V_{jet}) + u'\delta V_{jet}$ in red, non-linear perturbed state $u(V_{jet} + \delta V_{jet})$ in blue, for $\delta V_{jet} = V_{jet}/4$) and two sections.

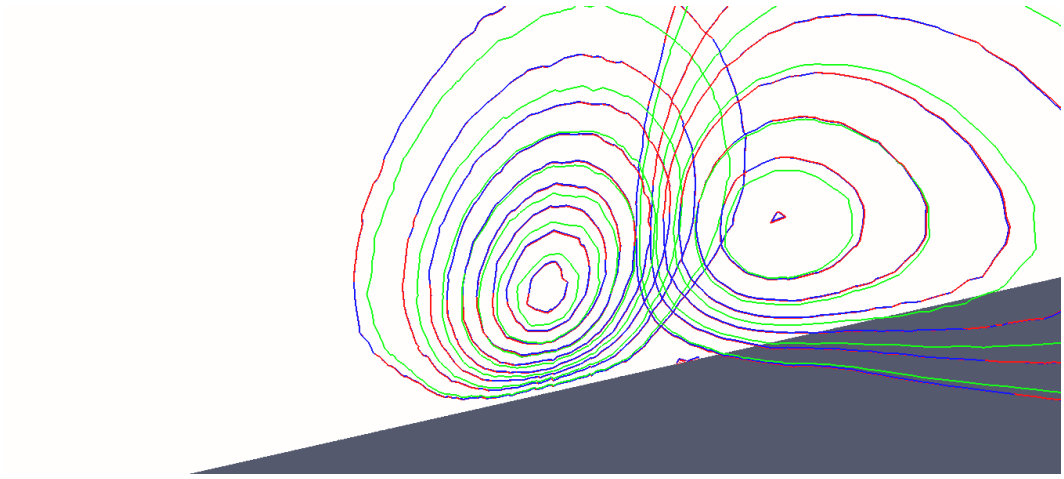


Figure 30: 3D jet case: linear extrapolation of crosswise velocity field w w.r.t. jet amplitude V_{jet} during blowing (reference state $w(V_{jet})$ in green, extrapolated state $w(V_{jet}) + w'\delta V_{jet}$ in red, non-linear perturbed state $w(V_{jet} + \delta V_{jet})$ in blue, for $\delta V_{jet} = V_{jet}/4$) and two sections.

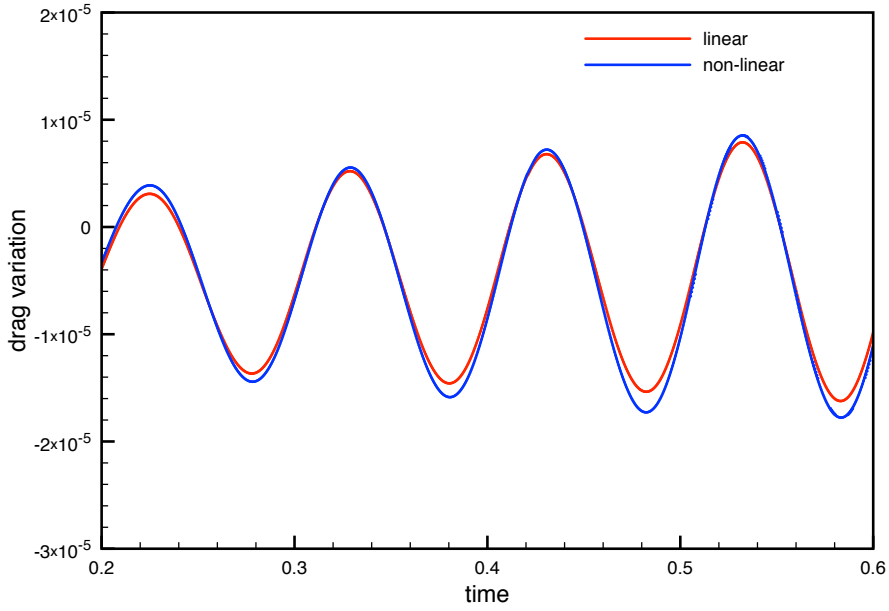


Figure 31: Jet case: comparison of time-dependent drag variation w.r.t. jet amplitude (non-linear variation $F_X(V_{jet}+\delta V_{jet})-F_X(V_{jet})$ in blue, linear variation $F'_X\delta V_{jet}$ in red, for $\delta V_{jet} = V_{jet}/4$).

Finally, some iso-surfaces of sensitivity fields are represented in Figs. (32 - 33). Sensitivity fields exhibit similar patterns as the 2D case, except a propagation in the crosswise direction, yielding spherical iso-surfaces in the vicinity of the jet.

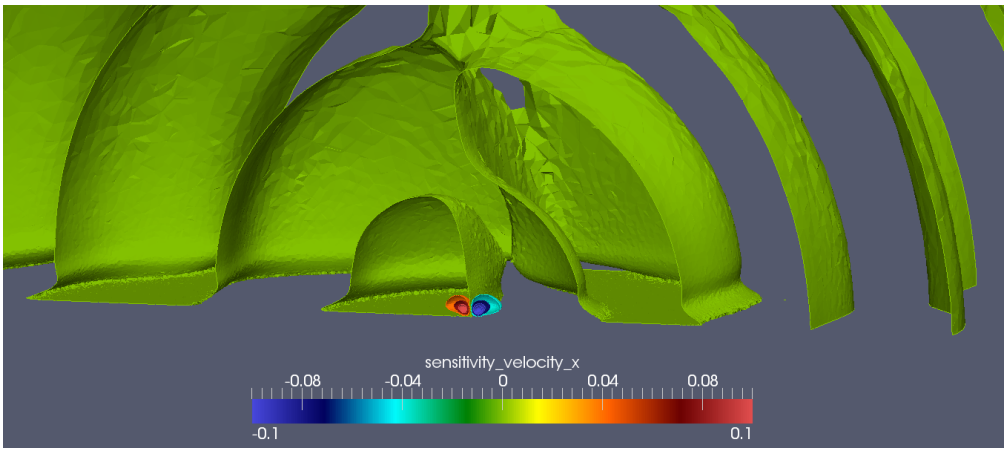


Figure 32: 3D jet case: some iso-surfaces of streamwise velocity sensitivity field u' during blowing.

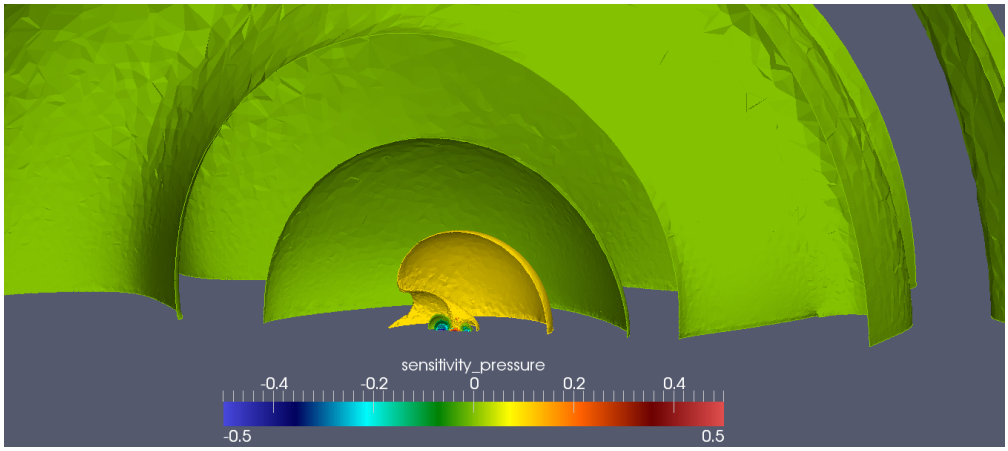


Figure 33: 3D jet case: some iso-surfaces of pressure sensitivity field p' during blowing.

7 Conclusion and prospects

A sensitivity equation method for unsteady compressible flows has been presented and verified on some test-cases corresponding to subsonic Eulerian and subsonic laminar viscous flows. The proposed approach yields a satisfactory computational efficiency, while being quite easy to implement in an existing code.

As prospects, we envisage now the extension to turbulent flows, which will require to solve the sensitivity of the turbulent quantities, as well as transonic problems, for which a modification of the Riemann solver for the sensitivities will be mandatory.

References

[1] P. Batten, N. Clarke, C. Lambert, and M. Causon. On the choice of wavespeeds for the hllc Riemann solver. *SIAM J. Sci. Comput.*, 18(6):1553–1570, November 1997.

[2] A. Belme. *Unsteady Aerodynamics and Adjoint method.* PhD thesis, Université de Nice Sophia-Antipolis, 2011.

[3] Jeff Borggaard and John Burns. A {PDE} sensitivity equation method for optimal aerodynamic design. *Journal of Computational Physics*, 136(2):366 – 384, 1997.

[4] A. Dervieux and J.-A. Désidéri. Compressible flow solvers using unstructured grids. INRIA Research Report 1732, June 1992.

[5] R. Duvigneau and D. Pelletier. On accurate boundary conditions for a shape sensitivity equation method. *Int. J. for Numerical Methods in Fluids*, 50(2), 2006.

[6] R. Duvigneau and D. Pelletier. A sensitivity equation method for fast evaluation of nearby flows and uncertainty analysis for shape parameters. *Int. J. of Computational Fluid Dynamics*, 20(7):497–512, August 2006.

[7] J. Elliott and J. Peraire. Practical 3d aerodynamic design and optimization using unstructured grids. *AIAA Journal*, 35(9):1479–1485, 1997.

[8] D. Ghate and M. B. Giles. Efficient Hessian Calculation Using Automatic Differentiation. In *Efficient Hessian Calculation Using Automatic Differentiation*, 2007.

[9] A. Griewank. Achieving logarithmic growth of temporal and spatial complexity in reverse automatic differentiation. *Optimization Methods and Software*, 1:35–54, 1992.

[10] V. Guinot, C. Delenne, and B. Cappelaere. An approximate Riemann solver for sensitivity equations with discontinuous solutions. *Advances in Water Resources*, 32, 2009.

[11] H. Hristova, S. Etienne, D. Pelletier, and J. Borggaard. A continuous sensitivity equation method for time-dependent incompressible laminar flows. *Int. J. for Numerical Methods in Fluids*, 50:817–844, 2004.

[12] A. Jameson. Aerodynamic design via control theory. *Journal of Scientific Computing*, 3(97-0101):233–260, 1988.

[13] A. Jameson, L. Martinelli, and N. A. Pierce. Optimum aerodynamic design using the Navier-Stokes equation. *Theorical and Computational Fluid Dynamics*, 10:213–237, 1998.

[14] T. Kloczko, C. Corre, and A. Beccantini. Low-cost implicit schemes for all-speed flows on unstructured meshes. *International Journal for Numerical Methods in Fluids*, 58(5):493–526, October 2008.

[15] N. Marco and A. Dervieux. Multilevel parametrization for aerodynamical optimization of 3d shapes. Technical report, INRIA Research Report No 2949, 1996.

[16] B. Mohammadi and O. Pironneau. Mesh adaptation and automatic differentiation in a cad-free framework for optimal shape. *International Journal for Numerical Methods in Engineering*, 30(2):127–136, May 1999.

[17] B. Mohammadi and O. Pironneau. *Applied Optimal Shape Design for Fluids*. Oxford University Press, 2001.

[18] E. J. Nielsen and W. K. Anderson. Aerodynamic design optimization on unstructured meshes using the Navier-Stockes equations. *AIAA Journal*, 37(11):1411–1419, 1999.

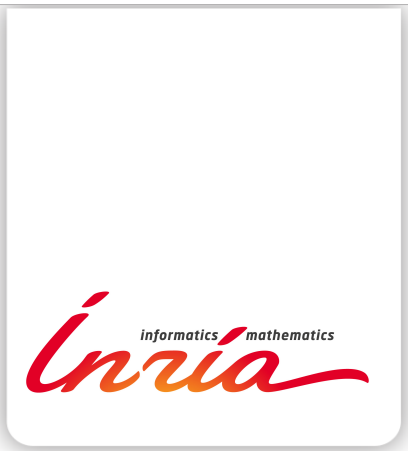
[19] L. L. Sherman, A. C. Taylor III, L. L. Green, and P. A. Newman. First and second-order aerodynamic sensitivity derivatives via automatic differentiation with incremental iterative methods. *J. Comput. Physics*, 129, 1996.

[20] E. Toro. *Riemann Solvers and Numerical Methods for Fluid Dynamics*. Springer Berlin Heidelberg, 1997.

[21] É. Turgeon, D. Pelletier, and J. Borggaard. Application of a sensitivity equation method to the $k - \epsilon$ model of turbulence. In *15th AIAA Computational Fluid Dynamics Conference*, Anaheim, CA, Jun. 2001. AIAA Paper 2001-2534.

[22] É. Turgeon, D. Pelletier, and J. Borggaard. Sensitivity and uncertainty analysis for variable property flows. In *39th AIAA Aerospace Sciences Meeting and Exhibit*, Reno, NV, Jan. 2001. AIAA Paper 2001-0139.

Contents



RESEARCH CENTRE
SOPHIA ANTIPOLIS – MÉDITERRANÉE

2004 route des Lucioles - BP 93
06902 Sophia Antipolis Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399